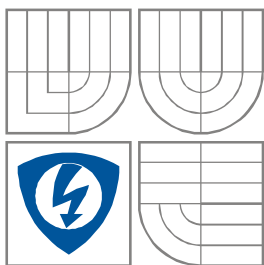


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## PARAMETRIZOVÁNÍ 2D ZAPISOVAČE

COMMANDS FOR 2D PLOTTER

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JAKUB HECZKO

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. RADEK ŠTOHL, PH.D.

BRNO 2012



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
**Automatizační a měřicí technika**

**Student:** Jakub Heczko

**ID:** 125439

**Ročník:** 3

**Akademický rok:** 2011/2012

## NÁZEV TÉMATU:

### Parametrizování 2D zapisovače

#### POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s instrumentací firmy Rockwell Automation pro model manipulátoru-zapisovače.
2. Realizujte aplikaci na PC, která bude úkolovat 2D model zapisovače pomocí moderních komunikačních prostředků.
3. Realizujte technické písmo do PLC (číslice).
4. Navrhněte a realizujte další možnosti ovládání 2D zapisovače.
5. Ověřte funkčnost systému.

#### DOPORUČENÁ LITERATURA:

Kinetix 2000 Multi-axis Servo Drive. (User Manual) Milwaukee: Rockwell Automation, Inc. 2007. 226 s.

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

**Termín zadání:** 6.2.2012

**Termín odevzdání:** 28.5.2012

**Vedoucí práce:** Ing. Radek Štohl, Ph.D.

**Konzultanti bakalářské práce:**

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

#### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Tato práce se zabývá parametrizováním modelu 2D zapisovače, jehož funkcí je vypisování znaků. Programy pro již vytvořená písmena jsou doplněny o znaky číslic. Dále jsou nastíněny možnosti vytvoření parametrizační aplikace pomocí MS Excel, vizualizačního prostředí FactoryTalkView a MS Word. V MS Word je pak celá aplikace realizována. Program pro ovládání, kontrolu a přenos znaků je řešen prostřednictvím VBA (Visual Basic for Applications). Data se mezi počítačem a řídicím systémem přenášejí přes DDE (Dynamic Data Exchange). V RSLogix 5000 je vytvořen program, který zadané znaky vypíše.

## **Klíčová slova**

2D zapisovač, parametrizace, VBA, DDE

## **Abstract**

This bachelor's thesis deals with commands for 2D plotter, which is used for writing characters. Programs for already created letters are supplemented by numeric digits. Possibilities of creating commanding application using MS Excel, FactoryTalkView software and MS Word are also mentioned. The final application is realized using MS Word. Program for controlling, verifying and transmission of characters is made by VBA (Visual Basic for Applications). Data between computer and PLC are transmitted via DDE (Dynamic Data Exchange). There is a program created in RSLogix 5000 which writes out the certain characters.

## **Keywords**

2D plotter, commands, VBA, DDE

### **Bibliografická citace:**

HECZKO, J. *Parametrizování 2D zapisovače*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 38s. Vedoucí bakalářské práce byl Ing. Radek Štohl, Ph.D.

## **Prohlášení**

„Prohlašuji, že svou bakalářskou práci na téma Parametrizování 2D zapisovače jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **25. května 2012**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce Ing. Radku Štohlovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **25. května 2012**

.....  
podpis autora

# Obsah

Úvod .....	9
1 Provedení a přístrojová skladba 2D zapisovače.....	10
2 Návrhy řešení .....	14
2.1 Aplikace pro parametrizování .....	14
2.1.1 MS Excel .....	14
2.1.2 Vizualizační prostředí .....	16
2.1.3 MS Word .....	17
3 Realizace .....	18
3.1 Vytvoření DDE serveru .....	18
3.2 Ukládání parametrů v PLC .....	20
3.3 Realizace v MS Word.....	21
3.3.1 Obsluha aplikace uživatelem .....	22
3.3.2 Vytvoření aplikace.....	23
3.3.3 Popis částí projektu a jejich podprogramů .....	24
3.3.4 Výpočet konstant .....	29
3.3.5 Přenos dat pomocí DDE .....	30
3.4 Realizace v RSLogix .....	30
3.4.1 Obsluha aplikace uživatelem .....	31
3.4.2 Popis podprogramů.....	31
4 Závěr .....	35
Literatura.....	36
Seznam příloh .....	38
A) PŘÍLOŽENÉ CD.....	38
B) UKÁZKOVÉ STRÁNKY .....	38

# Seznam obrázků

Obrázek 1.1: 2D Zapisovač .....	10
Obrázek 1.2: Rozložení pracovní plochy.....	11
Obrázek 1.3:GuardLogix, komunikační moduly, SERCOS Interface .....	12
Obrázek 1.4: Umístění indukčnostního senzoru a bezpečnostního koncového spínače .....	13
Obrázek 2.1: Aplikace v MS Excel 1 .....	15
Obrázek 2.2: Aplikace v MS Excel 2 .....	15
Obrázek 2.3: Aplikace v MS Excel 3 .....	16
Obrázek 2.4: Aplikace ve vizualizačním prostředí 1 .....	16
Obrázek 2.5: Aplikace ve vizualizačním prostředí 2 .....	17
Obrázek 2.6: Aplikace ve vizualizačním prostředí 3 .....	17
Obrázek 3.1: Export tagů .....	18
Obrázek 3.2: Výběr procesoru.....	19
Obrázek 3.3: Vytvoření nového Topic .....	19
Obrázek 3.4: Propojení s vyexportovanými tagy (symboly) .....	20
Obrázek 3.5: Pole poleznak o 1000 prvcích typu dattypznak.....	21
Obrázek 3.6: Ovládací formulář po ověření validity textu .....	22
Obrázek 3.7:Kalibrace fixu - info. okno.....	23
Obrázek 3.8: Vývojový diagram - kontrola jednoho znaku.....	26
Obrázek 3.9: Rozměry kuželky znaku vel. 550b .....	29
Obrázek 3.10: Vývojový diagram - zpracování znaku .....	32
Obrázek 3.11: Vývojový diagram - výběr znaku.....	33
Obrázek 3.12: Vzor technického písma - číslice .....	34



# ÚVOD

2D zapisovač je aplikace realizovaná na stroji portálového typu, který se pohybuje ve třech osách a umožňuje vypisování znaků. Horizontální osy vykonávají vykreslování znaků, vertikální slouží pouze ke spouštění psacího prostředku na plochu.

Úkolem této práce je parametrizování zapisovače, čímž se rozumí zadávání textu, vlastností znaků (velikosti a natočení) a následné provedení psací úlohy.

Práce navazuje na předchozí bakalářskou práci p. Poláka [1], jejímž úkolem bylo vytvořit pro zapisovač technické písmo (pouze znaky velké abecedy s diakritikou). Není tedy možné psát jakékoli znaky jakéhokoli fontu, ale jen ty, které jsou naprogramovány.

Pro lepší pochopení ovládání zapisovače je vhodné si prostudovat jak výše zmíněnou bakalářskou práci, tak i práci zabývající se řízením 3 os tohoto manipulátoru p. Matějčka [2], případně i práci o vizualizaci manuálního ovládání manipulátoru p. Chlada [3].

# 1 PROVEDENÍ A PŘÍSTROJOVÁ SKLADBA 2D ZAPISOVAČE

Konstrukčně se jedná o stroj portálového typu s rozměrem plochy zapisovače 780x1080mm. Jako nástroj slouží psací fix.



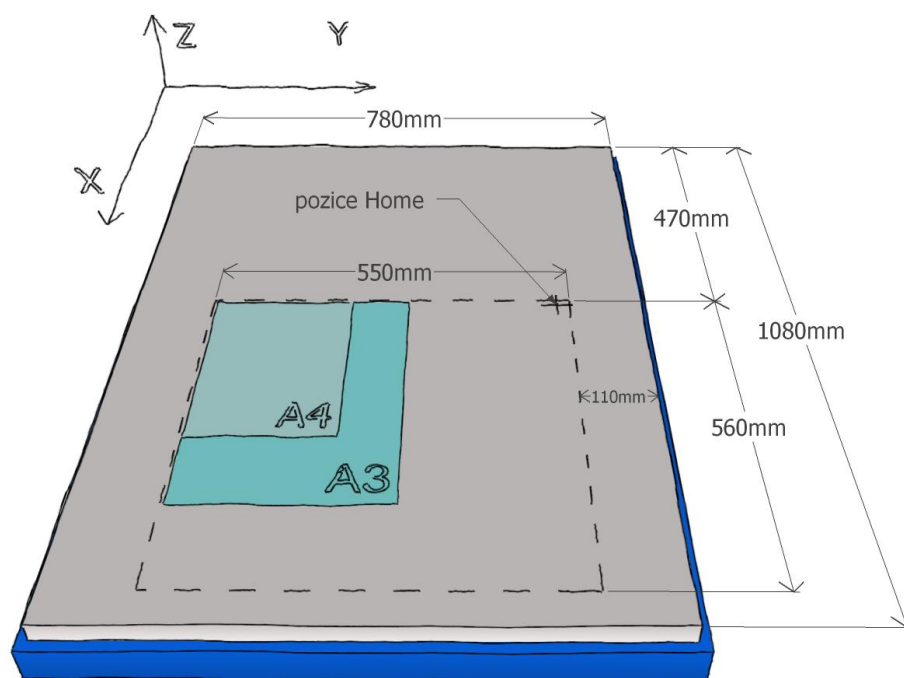
Obrázek 1.1: 2D Zapisovač

## Rozložení pracovní plochy

Při inicializaci zapisovač najede do pozice Home a na všech osách se nastaví pozice nula (0,0,0). Oblast, ve které je možné se pohybovat, je omezena parametry stroje. Aby se osy nedostaly za tyto meze a nezpůsobily kolizi, jsou hlídány softwarovými i hardwarovými limity. Softwarové limity jsou:

- osa X            -10, +550 mm
- osa Y            -20, +530 mm
- osa Z            -20, +120 mm

Výsledná pracovní plocha, ve které se fix může pohybovat, je 560x550 mm.



Obrázek 1.2: Rozložení pracovní plochy

Pozice Home je vpravo nahoře, avšak papír je podle zvyklostí umístěn vlevo nahoře. Abychom měli v levém horním rohu papíru pozici (0,0), musíme nastavit příslušné off-sets. Na ose Y je to 525mm (530mm – 5mm jako rezerva před softwarovým limitem). Na ose X pak pouze ona rezerva 5mm.

Reálné pozice předávané zapisovači jsou:

osa X = vypočtená pozice znaku na papíře – 5

osa Y = 525 – vypočtená pozice znaku na papíře

## Popis komponentů

Každá osa má svůj pohon. Osy pro horizontální pohyb jsou poháněny servomotory MPL A310F HK22AA. Osa Z, která zvedá a spouští fix, je připojena na servomotor MPL A310 HK24AA. Tento motor je navíc vybaven brzdou. Oba typy motorů jsou napájeny 230V, mají maximálně 3000 ot./min a zpětnou vazbu dává vestavěný enkodér.

Servopohony Kinetix 2000 [4] umožňují řízení servomotorů. Využívají řídicí platformu Logix, například ControlLogix. Pro programování pohonů i procesů se používá vývo-

jový systém RSLogix 5000. To umožňuje řídit motory, diskrétní události a jiné v rámci jednoho programu.

Pro komunikaci mezi PLC a Kinetix 2000 se využívá vysokorychlostního modulu SERCOS Interface1756-M08SE [5] schopného obsluhovat až 8 pohonů. Data se přenášejí po optických vláknech, v tomto případě po plastových optických kabelech 2090-SCEP3-0. Přenosová rychlost je 4 nebo 8 Mbps a udává dobu cyklu 1 ms nebo 0,5 ms.

Jako PLC je použit automat třídy GuardLogix [6]. Je to řídicí systém ControlLogix, který poskytuje safety řízení. Obsahuje dva procesory. Primární procesor je 1756-L62S a druhý 1756-LPS je jeho safety partner. Oba běží paralelně, avšak bezpečnostní funkce jsou nadřazeny standardním. Bezpečnostní procesor obsahuje safety program, který je po naprogramování uzamčen, a tím je chráněn před případnou nežádanou změnou. Výhoda tohoto systému spočívá v tom, že spojuje standardní funkce ControlLogix, jako je řízení procesů, pohonů a polohování a safety část do jednoho řídicího celku. Pro programování a komunikaci s počítačem se využívá EtherNet/IP modul 1756-ENBT. Modul DeviceNet 1756-DNB slouží ke komunikaci se Safety I/O modulem.

Safety I/O modul 1971DS-IB8XOB8 obsahuje 8 digitálních vstupů a 8 digitálních výstupů. Připojují se do něj bezpečnostní prvky, např. optické závory, stop tlačítka, koncové spínače apod.



Obrázek 1.3: GuardLogix, komunikační moduly, SERCOS Interface

## Bezpečnostní prvky

O bezpečnost celého stroje se stará safety část PLC GuardLogix a bezpečnostní prvky, jako jsou optická závora, bezpečnostní tlačítko a koncové spínače. Ty jsou připojeny do safety I/O modulu.

Optická závora (typ: 440L-T4J0480YD) je umístěna v manipulačním prostoru zapisovače (Obrázek 1.1). Chrání obsluhu, zatímco je stroj v chodu.

Bezpečnostním tlačítkem se vypíná stroj v případě nouze. Je přimontováno na rozvaděči manipulátoru.

Koncové spínače (typ: 440P-C 22mm) fungují jako hardwarové limity. Jsou umístěny na obou koncích každé osy. Představují poslední stupeň ochrany před vyjetím zařízení z osy. Mimo tuto ochranu jsou definovány softwarové limity, které by neměly být překročeny a vymezují správnou maximální pracovní plochu zapisovače. Funkce Homing využívá indukční senzory Pulsotronic, které jsou umístěny na jednom kraji os. Při najetí do pozice Home je poloha vynulována.



Obrázek 1.4: Umístění indukčního senzoru a bezpečnostního koncového spínače

## 2 NÁVRHY ŘEŠENÍ

Abychom mohli 2D zapisovač parametrizovat, je nutné vymyslet aplikaci, ve které se budou ony parametry zadávat. S tím souvisí také vyřešení toho, jakým způsobem se budou přenášet data mezi aplikací a PLC a jak se budou parametry v PLC ukládat.

### 2.1 Aplikace pro parametrizování

V aplikaci musí být možné určitým způsobem zadávat znaky a jejich vlastnosti.

Možnosti, jak je zadávat, jsou například tyto:

1. Pokud chceme, aby znaky v jednom řetězci mohly mít různou velikost, popř. natočení, můžeme zapisovat znak po znaku a ke každému znaku přiřazovat různé vlastnosti.
2. Další možností je, že celý řetězec bude mít stejné vlastnosti. Napíšeme text a jeden údaj o velikosti a natočení. Tyto vlastnosti se pak nakopírují ke všem prvkům pole.
3. Poslední možnost představuje psaní textu v definovaném formátu, ze kterého bude možné určit vlastnosti znaků. Například na začátku věty je ve značce zapsán údaj o vlastnostech celého následujícího řetězce. Tyto vlastnosti se nemění až do další značky, která definuje vlastnosti následující věty. Značka je volena taková, u které se nepředpokládá její běžné použití v textu. Může mít tvar (\\###,###). První číslo určuje velikost, druhé natočení.

Prostředí pro vytvoření aplikace se nabízí několik. Můžeme použít vizualizační prostředí FactoryTalkView nebo tabulkový procesor MS Excel. Avšak zhodnocením vlastností předchozích prostředí se začal jevit vhodným i textový editor MS Word.

#### 2.1.1 MS Excel

Součástí MS Excel je VBA (Visual Basic for Application), což umožňuje vytvářet jednoduché skripty potřebné pro zpracování dat a vytvoření komunikace. MS Excel podporuje výměnu dat pomocí DDE (Dynamic Data Exchange) [7]. Jako DDE klient

slouží MS Excel a jako DDE server použijeme software RSLinx. RSLinx umí vytvořit i OPC server [8], přes který lze komunikovat také, avšak je nutné použít další software, který bude představovat OPC klienta. Teprve ten předává data MS Excelu. Jako OPC klienta můžeme využít například OPCEx – OPC Add-in Klient pro MS Excel od firmy Foxon [9]. Postup vytvoření DDE serveru v softwaru RSLinx viz *Vytvoření DDE serveru*.

### Ukázky aplikací

Vytvořil jsem tři ukázky parametrizačních aplikací, podle výše zmíněných možností zadávání parametrů. Každá aplikace musí obsahovat alespoň tlačítko pro odeslání dat do řídicího systému.

1. Tento způsob zadávání je poněkud pracnější. Hodí se, pokud chceme, aby každý znak měl jiné vlastnosti.

	A	B	C	D	E	F
1						
2		ZNAK	VELIKOST [CM]	NATOČENÍ [°]		Zapsat
3		V	20	0		
4		U	20	0		
5		T	20	0		
6			15	0		
7		V	15	0		
8			15	0		
9		B	15	0		
10		R	15	0		
11		N	15	0		
12		Ě	15	0		
13						
14						

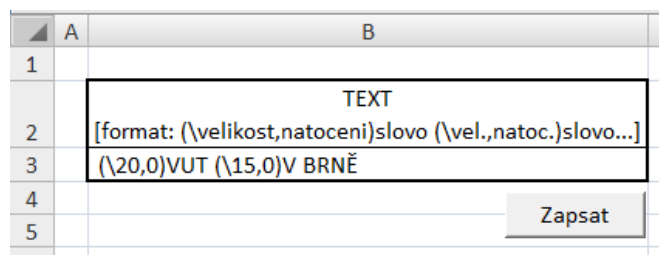
Obrázek 2.1: Aplikace v MS Excel 1

2. Tato možnost je vhodná, máme-li v úmyslu psát delší texty s konstantními vlastnostmi.

	A	B	C	D
1				
2		TEXT	VELIKOST [CM]	NATOČENÍ [°]
3		VUT V BRNĚ	20	0
4				
5				Zapsat

Obrázek 2.2: Aplikace v MS Excel 2

3. Třetí možnost umožňuje relativně snadno zadávat i delší text, navíc oproti druhé možnosti lze měnit vlastnosti znaků i v průběhu řetězce. Tento ukázkový příklad vypíše text se stejnými vlastnostmi, jako příklad 1.



Obrázek 2.3: Aplikace v MS Excel 3

## 2.1.2 Vizualizační prostředí

Použití firemního vizualizačního prostředí FactoryTalkView MachineEdition [10] má tu výhodu, že s řídicím systémem komunikuje přímo, pouze se spojením vytvořeným v RSLinx. Není tedy nutné využívat OPC, DDE nebo nějakým jiným způsobem vytvářet komunikaci.

### Ukázky aplikací

Principy zadávání jsou stejné jako u MS Excel. Liší se pouze vzhledem, proto uvádím pouze obrázky bez popisu.

1.



Obrázek 2.4: Aplikace ve vizualizačním prostředí 1



2.

Text	Velikost	Natoceni
VUT V BRNĚ	20	0
		Zapis

Obrázek 2.5: Aplikace ve vizualizačním prostředí 2

3.

Text [format: (velikost,natoceni)slovo(wel.,natoc.) slovo...]
(\20,0)VUT(\15,0) V BRNĚ
Zapis

Obrázek 2.6: Aplikace ve vizualizačním prostředí 3

### 2.1.3 MS Word

MS Word stejně jako MS Excel obsahuje VBA a umožňuje výměnu dat pomocí DDE. Navíc má tu výhodu, že text, který chceme vypsát zapisovačem, píšeme jako běžný dokument. Stejně tak mu zadáváme i velikost. Dalším plusem je, že text lze na stránce jednoduše umístit kamkoliv. To by u předchozích možností nebylo tak jednoduše proveditelné. Nevýhodou však je, že text nelze natáčet, MS Word takovou možnost neumožňuje.

Usoudil jsem, že možnost umístit text kamkoli na stránku i jeho přirozené psaní je přínosnější, než natočení. Proto jsem celé parametrizování 2D zapisovače řešil pomocí MS Word.

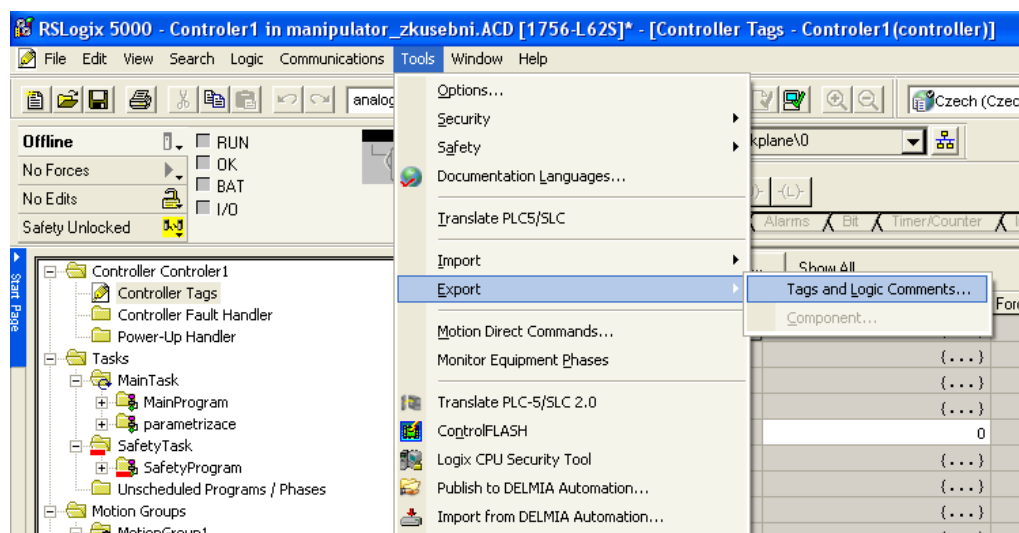
## 3 REALIZACE

Aplikaci pro zadávání textu, jeho zpracování a řízení přenosu dat jsem z výše uvedených důvodů realizoval pomocí MS Word. Dále jsem v RSLogix vytvořil program, který obsluhuje 2D zapisovač a vypisuje námi zadané znaky. Pro přenos dat slouží DDE. Dalším bodem zadání je rozšíření vytvořených znaků písmen o znaky číslic.

### 3.1 Vytvoření DDE serveru

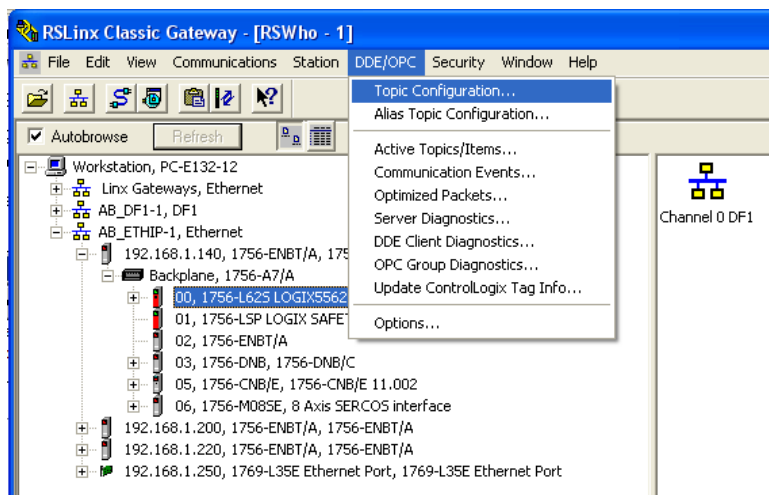
Abychom se mohli připojit s klientem na server, musíme ho nejprve nastavit [11]. Jako DDE server použijeme RSLinx Classic.

1. V prvním kroku si z RSLogix 5000 exportujeme tagy do souboru, který budeme načítat při nastavování serveru. Vybereme položku **Tools - Export - Tags and Logic Comments** a soubor uložíme.



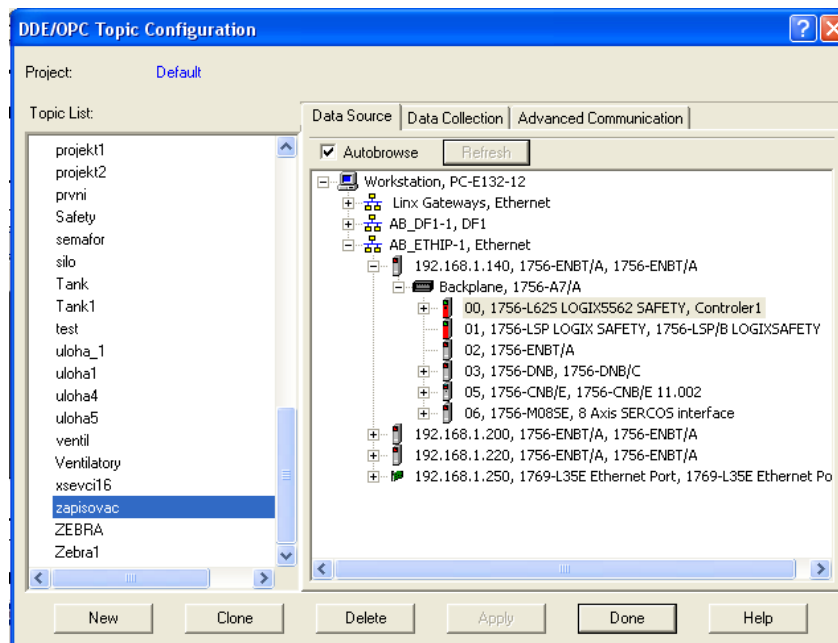
Obrázek 3.1: Export tagů

2. V RSLinx vybereme processor, ve kterém máme nahraný program a klikneme na **DDE/OPC – Topic Configuration**



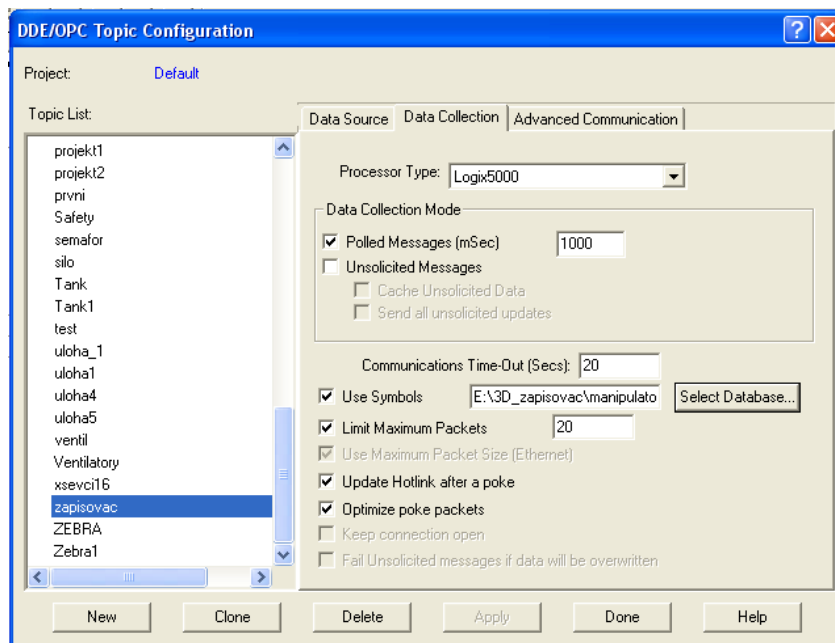
Obrázek 3.2: Výběr procesoru

3. Nyní vytvoříme nový Topic kliknutím na **New** a pojmenujeme ho například „zapisovac2D“. V pravé části okna se ujistíme, že jsme vybrali správný procesor.



Obrázek 3.3: Vytvoření nového Topic

4. Na záložce **Data Collection** zaškrtneme **Use Symbols** a vybereme námi předem vyexportovaný soubor tagů.



Obrázek 3.4: Propojení s vyexportovanými tagy (symboly)

Potvrdíme a tímto je hotova konfigurace DDE serveru.

## 3.2 Ukládání parametrů v PLC

Jelikož budeme vytvářet aplikaci v MS Word, potřebujeme ukládat pouze údaj o znaku, jeho velikosti a umístění na papíře. Umístění je dáno vzdáleností počátku znaku od horní a levé hrany papíru. V programu RSLogix 5000 [12] si vytvoříme uživatelem definovaný typ `dattypznak` pomocí struktury. Ten bude obsahovat znak (DINT), velikost (REAL), `poziceshora` (REAL) a `pozicezleva` (REAL). Do znak se ukládá ASCII kód znaku, velikost obsahuje výšku znaku v milimetrech. `poziceshora` a `pozicezleva` jsou dány v milimetrech a společně určují levý dolní roh znaku.

Dále vytvoříme pole proměnných `poleznak` datového typu `dattypznak`. Počet prvků pole určuje, kolik lze maximálně vypsát znaků. Já jsem zvolil délku pole 1000.

	Name	Value	Force Mask	Style	Data Type	Class
	poleznak	{...}	{...}		datypznak[1000]	Standard
	poleznak[0]	{...}	{...}		datypznak	Standard
	poleznak[0].znak	0		Decimal	DINT	Standard
	poleznak[0].velikost	0.0		Float	REAL	Standard
	poleznak[0].poziceshora	0.0		Float	REAL	Standard
	poleznak[0].pozicezleva	0.0		Float	REAL	Standard
	poleznak[1]	{...}	{...}		datypznak	Standard
	poleznak[2]	{...}	{...}		datypznak	Standard
	poleznak[3]	{...}	{...}		datypznak	Standard
	poleznak[4]	{...}	{...}		datypznak	Standard

Obrázek 3.5: Pole poleznak o 1000 prvcích typu datypznak

### 3.3 Realizace v MS Word

Text se zadává přímo do textového editoru. K ovládání slouží vytvořený ovládací formulář, který je spolu se zpracováním a přenosem dat řešen pomocí VBA. Použitý font pro zadávání textu se nazývá *osifont*<sup>1</sup>. Jedná se o technické písmo pro technické výkresy, takže koresponduje s písmem vytvořeným v zapisovači.

**Ukázka *osifont***

0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ

ÁČĎÉĚĪŇÓŘŠŤÚŮÝŽ

<sup>1</sup> Font je možno stáhnout na adrese <http://ponkrac.net/pisma/osifont> [20.5.2012]

### 3.3.1 Obsluha aplikace uživatelem

Po otevření dokumentu s uloženým programem se spustí ovládací formulář, viz Obrázek 3.6.

Parametrizace 2D zapisovače

Ověření validity textu      Ruční kalibrace fixu

Vypsát text zapisovačem      Konec

[6]      Znaky se budou vypisovat na papír velikosti 297 x 210mm (A4 na výšku).  
Pokud je vložen správný papír, můžete pokračovat vypsáním textu zapisovačem.

[5]      Počet znaků k vypsání: 40

[4]      Zadaný text je v pořádku.

[3]      Kontrola validity textu...

[2]      Zadejte text k vypsání zapisovačem. Mohou to být pouze písmena velké abecedy  
včetně diakritiky a číslice. Poté pokračujte ověřením validity textu a výpisem textu  
zapisovačem.

[1]      Parametrizace 2D zapisovače  
Jakub Heczko  
Bakalářská práce 2012

Obrázek 3.6: Ovládací formulář po ověření validity textu

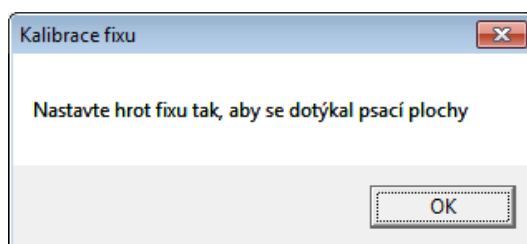
Nyní lze začít psát text do textového editoru. Povolené znaky jsou ty, které jsou vytvořeny v programu pro řídicí systém. Jsou to písmena velké abecedy, včetně diakritiky a číslice. S textem lze provádět standardní úpravy, které se však mohou týkat pouze jeho rozložení na stránce. To znamená měnit řádkování, odsazení, zarovnání, mezery apod. Dále je možné nastavovat velikost stránky, orientaci, okraje a další. Nelze ovšem měnit proporce písma, jako je font, tučné zvýraznění, kurzíva atd. To by mělo za následek nekompatibilitu s vytvořenými znaky v řídicím systému a mohlo by to vést např. k psaní mimo papír. Není také možné rozložit text na více než jednu stranu.

Pokud je text dokončen, lze přistoupit k **Ověření validity textu**. Tím se zkontroluje, zda text neobsahuje nepovolené znaky nebo úpravy. V případě, že ano, označí se tyto znaky červeně a nelze pokračovat, dokud nejsou chyby opraveny.

Jestliže je vše v pořádku, v ovládacím formuláři v textovém okně se zobrazí počet znaků k vypsání, informace o velikosti použitého papíru, o jeho orientaci a o tom, že je možné přejít k vypsání textu zapisovačem viz Obrázek 3.6: Ovládací formulář po ověření validity textu.

V případě, že je text dopsán a dojde ke zmáčknutí tlačítka **Vypsat text zapisovačem**, data se přenesou z počítače do řídicího systému a zapisovač začne vypisovat znaky.

**Ruční kalibrace fixu** umožňuje nastavit hrot fixu na psací plochu. Zmáčknutím tohoto tlačítka dojde ke spuštění osy s fixem do polohy, ve které bude následně psát, a také se zároveň zobrazí informační okno s potvrzením. V tomto kroku je nutné ručně vyladit pozici hrotu fixu tak, aby přesně doléhal na papír. Toto nastavení fixu se provádí upínacím šroubem. Po seřízení se informační okno potvrdí a osa se vrátí zpět do výchozí polohy.



Obrázek 3.7: Kalibrace fixu - info. okno

Tlačítko **Konec** nabízí možnost uložení dokumentu a následně jeho ukončení.

Textové okno v ovládacím formuláři slouží k informování o stavu programu a také radí, jak dále postupovat.

### 3.3.2 Vytvoření aplikace

Aplikaci budeme vytvářet v prostředí MS Word 2007.

Otevřeme si nový dokument v MS Word. Defaultně není zobrazena karta **Vývojář**. Její zobrazení provedeme kliknutím na **Tlačítko Microsoft Office**, dále **Možnosti aplikace Word**. Na kartě **Oblíbené** zaškrtneme **Zobrazit na pásu kartu Vývojář**. V kartě **Vývojář** klikneme na ikonu **Visual Basic** a tím se dostaneme do vývojového prostředí. Lze použít i klávesovou zkratku Alt+F11.

Vlevo v okně **Project** se nyní nachází složka **Objects** a v ní **ThisDocument**. My si ještě vytvoříme ovládací formulář a modul pro psaní kódu. To provedeme kliknutím pravým tlačítkem na **Project**, **Insert**, **UserForm** a pak **Module**. Vytvořený Module1 si přejmenujeme na ZpracovaniTextu.

### 3.3.3 Popis částí projektu a jejich podprogramů

#### **ThisDocument**

`Document_Open`

Tento podprogram se provede ihned po otevření dokumentu. Má za úkol nastavit defaultní vlastnosti fontu. Především font *osifont*, netučné, bez kurzivy a všechny znaky velké. Dále zobrazí vytvořený formulář *UserForm1*.

#### **UserForm1**

Formulář *UserForm1* osadíme tlačítky *CommandButton* a textovým oknem *TextBox*. Lze je vybrat z nabídky *Toolbox*. Následně se přepneme do zobrazení kódu, kde se vytvářejí podprogramy navázané na události týkající se *UserForm*.

`btnKalibraceFixu_Click`

Nastaví bit `kalibrace_fixu` v PLC na log. 1. Tím dá povel, aby zapisovač nastavil osu do polohy pro kalibraci. Zobrazí se vyskakovací okno, které uživateli dá pokyn, co má udělat. Až uživatel fyzicky provede kalibraci a potvrdí vyskakovací okno, bit `kalibrace_fixu` se nastaví na log. 0.

`btnKonec_Click`

Pokud byl dokument již uložen, ukončí ho. Jestliže ne, dá uživateli možnost ho uložit.

`btnValidita_Click`

Spustí podprogram `KontrolaZnaku`.

`btnVypsati_Click`

Spustí podprogram `ZpracovaniTextuu`.

`TextBox1_Change`

Jestliže dojde ke změně textového okna *TextBox1* (např. výpisu textu), nastaví se jako aktivní první řádek. V praxi to znamená, že pokud je v textovém okně více textu, než lze zobrazit bez scrollování, horní řádek zůstává stále nahoře.



UserForm\_Initialize

Spustí se při inicializaci. Nastaví *TextBox1* jako víceřádkový, zamknutý proti upravování, aktivní, se *ScrollBar* a také ho vyprázdní. Dále se nastaví *btnVypsat* jako neaktivní, index vypisování *tbIndex* se vynuluje a pomocí podprogramu *tbVypis* se do *TextBox1* vypíše úvodní slova.

## ZpracovaniTextu

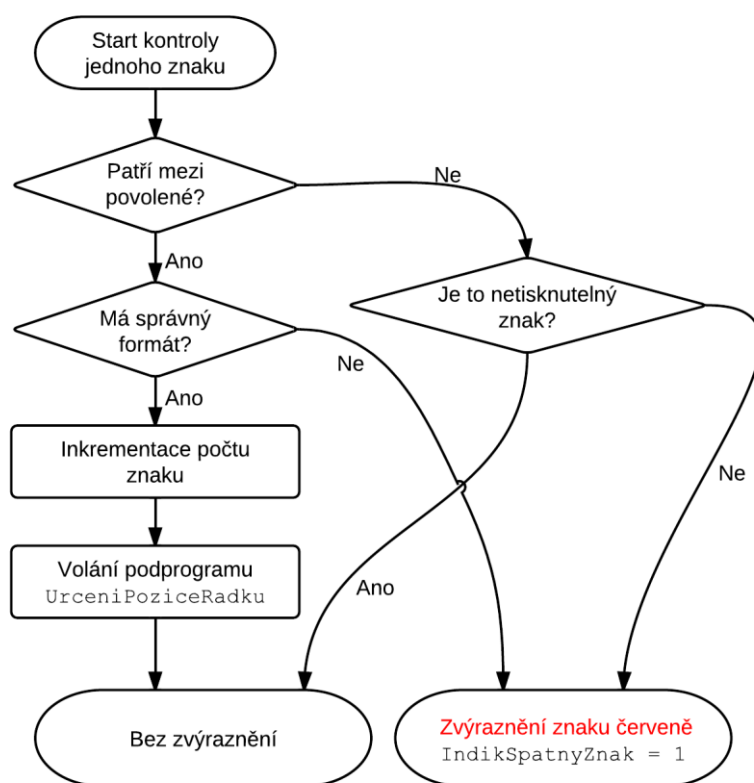
V modulu *ZpracovaniTextu* se kontroluje validita textu, dochází k jeho zpracování a odesílání do PLC.

KontrolaZnaku

Má za úkol ověřit, jestli je zadaný text ve správném formátu, jestli neobsahuje nepovolené znaky a také, jestli dokument nemá více, než jednu stránku.

Jako první se provede kontrola počtu stránek. Pokud jich je více než jedna, dojde k informování uživatele a *KontrolaZnaku* se ukončí.

Text je kontrolován znak po znaku. Nejdříve se ověří, zda znak náleží mezi A-Z 0-9 ÁČĎÉĚĪŇÓŘŠŤÚŮÝŽ. V případě, že ano, zkontroluje se ještě jeho formát (font, tučné, kurzíva). Znak může také náležet do netisknutelných znaků (mezera, tabulátor, zalomení řádku), toto se testuje v dalším kroku. Pokud se zjistí, že znak nevyhovuje některé z podmínek, označí se červeným zvýrazněním a nastaví se ukazatel *IndikSpatnyZnak*. Ostatní povolené znaky zůstanou nezvýrazněné. Počet tisknutelných znaků se ukládá do proměnné *PocetSpravnychZnaku*. V průběhu kontroly se ještě volá podprogram *UrceniPoziceRadku(index)*.



Obrázek 3.8: Vývojový diagram - kontrola jednoho znaku

Po kontrole všech znaků nastává vyhodnocení. Pokud byl zadán špatný znak ( $\text{IndikSpatnyZnak}=1$ ), nebo nebyl zadán žádný ( $\text{PocetSpravnychZnaku}=0$ ), případně bylo-li zadáno více znaků, než je možné uložit v PLC ( $\text{PocetSpravnychZnaku}>\text{MaxPocetZnaku}$ ), dojde k informování uživatele přes zprávu v *TextBoxu* a *btnVypsát* zůstane neaktivní. To zamezí pokračování na podprogram *ZpracovaniTextuu*, dokud uživatel nezadá všechny znaky korektně.

Jsou-li všechny znaky zadány správně, uživatel je informován, že kontrola proběhla v pořádku, je mu sděleno, kolik znaků bude vypsáno, provede se podprogram *InfoVelikostPapiru* a dojde k aktivování tlačítka *btnVypsát* pro pokračování programu.

#### ZpracovaniTextuu

Pokud *KontrolaZnaku* proběhla úspěšně, tlačítko *btnVypsát* se stalo aktivní a dává možnost spustit tento podprogram.

Ověří se, jestli od doby, kdy byla provedena `KontrolaZnaku`, nedošlo ke změně textu. V případě, že ano, je uživatel informován a tlačítko `btnVypsati` se opět stane neaktivním.

Text je znovu procházen znak po znaku. Pokud znak náleží mezi povolené (A-Z0-9ÁČĎĚĚĚŇÓŘŠŤÚŮÝŽ), naplní se struktura prvku pole `poleZnak`, což probíhá následovně.

Do `Znak` se uloží ASCII hodnota znaku.

`Velikost` je závislá na velikosti znaku. Pro přepočítání na mm se musí velikost znaku v bodech vynásobit konstantou `KonstVelPisma`.

`PoziceZleva` udává vzdálenost od levého okraje papíru k počátku znaku. Je potřeba k ní připočítat velikost znaku v bodech vynásobenou konstantou `KonstPoziceZleva`. Jelikož má zapisovač pozici 0 na ose Y vpravo a tato pozice roste směrem doleva, je nutné vypočtenou polohu znaku odečíst od čísla 525. Tímto krokem docílíme přesunu počátku psaní na levou stranu plochy, 5 mm od softwarových limitů.

Do `PoziceShora` se uloží vzdálenost od horního okraje papíru po počáteční dolní roh znaku s připočtenými 5mm pro rezervu před softwarovým limitem. Znaky na jednom řádku musí mít tuto hodnotu stejnou. Proto se k jejímu určení využívá hodnoty `PoziceRadku`, získané v podprogramu `UrceniPoziceRadku(index)`.

Postup výpočtu konstant a pozice znaku viz Výpočet konstant.

Jestliže znak nesplňuje podmínku, tj. nepatří mezi povolené znaky, jedná se o znak netisknutelný. V tomto případě inkrementujeme proměnnou `pomZarovnani`, která upravuje uspořádání v poli `poleZnak`.

Po uložení všech znaků do `poleZnak` v počítači přeneseme do PLC údaj o množství odesílaných znaků `PocetSpravnychZnaku`, a to pomocí podprogramu `PrenosDoRSLogix(co, kde)`.

Dále do PLC odešleme celé pole `poleznak`. Pro přesun využijeme podprogram `PrenosZnaku`.

Jakmile budou všechny znaky přesunuty, nastaví se v PLC proměnná `param_start`. Tím se dá zapisovači pokyn, aby začal provádět program pro výpis znaků. Nastavení bitu provedeme opět pomocí podprogramu `PrenosDoRSLogix(co, kde)`.

`PrenosZnaku`

Přenesení celé naplněné pole `poleznak` z počítače do PLC. Viz Přenos dat pomocí DDE.

`PrenosDoRSLogix(co, kde)`

Slouží pro přenesení jedné proměnné z počítače do PLC. Parametr `co` je pro proměnnou, kterou chceme přenést, a parametr `kde` znamená, pod jakou proměnnou ji hodláme uložit v PLC.

`PrenosZRSLogix(co, kde)`

Přenáší proměnnou z PLC do počítače. Způsob je obdobný jako u `PrenosDoRSLogix(co, kde)`.

`tbVypis(text)`

Vypíše parametr `text` do *TextBoxu* v *UserForm*. Využívá se k informování uživatele o stavu programu a také mu dává instrukce jak pokračovat. Vložený text se přidá před ostatní a uvádí ho index.

`InfoVelikostPapiru`

Po provedení se pomocí `tbVypis` vypíše informace o rozměrech papíru v mm a pro upřesnění i jeho orientace (na výšku, na šířku). Pokud se jedná o standardní formát, připojí se jeho název (A5, A4, A3).

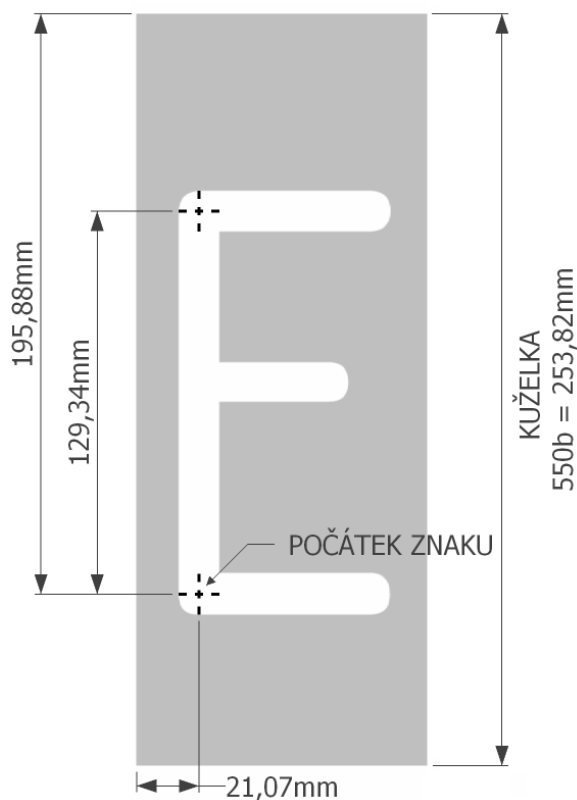
`UrceniPoziceRadku(index)`

Postupným voláním z podprogramu `KontrolaZnaku`, projde všechny znaky na jednom řádku a podle největšího z nich určí pozici pro všechny znaky nacházející

se na stejném řádku. Vzdálenost od horního okraje papíru se ukládá do pole `PoziceRadku`.

### 3.3.4 Výpočet konstant

Výška znaku je z MS Word dána v bodech. Pro přepočet na centimetry (pro zapisovač) je nutné vynásobit ji konstantou (`KonstVelPisma` (1)). MS Word také umožňuje zjistit vzdálenost znaku z horního i z levého okraje papíru. Tyto vzdálenosti jsou však brány od hrany papíru, po okraj kuželky znaku (šedé ohraničení kolem znaku na Obrázek 3.9). Abychom je přepočítali na reálný počátek znaku (levý dolní roh), musíme je rovněž vynásobit konstantami (`KonstPoziceShora` (2), `KonstPoziceZleva` (3)).



Obrázek 3.9: Rozměry kuželky znaku vel. 550b

$$\text{KonstVelPisma} = \frac{\text{reálna velikost znaku (cm)}}{\text{velikost kuželky v bodech}} = \frac{12,934}{550} = 0,0235 \text{ cm/b} \quad (1)$$

$$\text{KonstPoziceShora} = \frac{\text{vzdálenost od horního okraje kuželky po poč.znaku (mm)}}{\text{velikost kuželky v bodech}} = \frac{195,88}{550} = 0,3561 \text{ mm/b} \quad (2)$$

$$\text{KonstPoziceZleva} = \frac{\text{vzdálenost od levého okraje kuželky po poč.znaku (mm)}}{\text{velikost kuželky v bodech}} = \frac{21,07}{550} = 0,0383 \text{ mm/b} \quad (3)$$

### 3.3.5 Přenos dat pomocí DDE

Následující ukázka předvádí základní principy použití příkazů pro vytvoření spojení se serverem, přenosem dat ze serveru do klienta a naopak [11].

```
Private Function OpenRSLinx() 'funkce pro vytvoreni spojeni s RSLinx
    OpenRSLinx = Application.DDEInitiate("RSLINX", "zapisovac2D")
End Function

Private Sub CommandButton1_Click() 'po stisku Button1 data z PLC
    rslinx = OpenRSLinx() 'Otevre spojeni s RSLinx
    znak = DDERequest(rslinx, "ABECEDA[8]") 'nacte hodnotu z promenne
    ABECEDA[8] v PLC a ulozi ji do promenne znak v pocitaci
    DDETerminate rslinx 'ukonci spojeni s RSLinx
End Sub

Private Sub CommandButton2_Click() 'udalost po stisku Button2
    rslinx = OpenRSLinx() 'Otevre spojeni s RSLinx
    DDEPoke rslinx, "ABECEDA[9]", znak ' zapis hodnoty z promenne
    znak v PC, do promenne ABECEDA[9] v PLC
    DDETerminate rslinx 'ukonci spojeni s RSLinx
End Sub
```

## 3.4 Realizace v RSLogix

Můj program vychází z bakalářské práce p. Poláka [1]. Využívá podprogramu automatika, který jsem upravil a přejmenoval na Parametrizace2D\_main, dále podprogramů na vypisování jednotlivých znaků abecedy (také upravených) a SafetyProgramu. Já jsem nově vytvořil podprogramy pro vypisování číslic (0-9), Parametrizace2D\_vyber\_znaku, Parametrizace2D\_zpracovani\_znaku.

### 3.4.1 Obsluha aplikace uživatelem

Po zapnutí napájení zapisovače je nutné vyčkat, než proběhne kalibrace os (svítí LED č. 0). Jakmile je kalibrace dokončena (LED č. 0 zhasne), tlačítkem č. 0 provedeme inicializaci, tj. osy najedou do pozice Home, kde se jejich poloha vynuluje, a následně se přesunou do výchozí pozice pro psaní. Nyní je zapisovač připraven a čeká, až uživatel v ovládacím formuláři zmáčkne tlačítko **Vypsat text zapisovačem**. Až zapisovač dopíše, vrátí se zpět do výchozí pozice a je připraven na další příkazy. Obsluha a signalizace chyb, případně další ovládání je stejné, jako ve výše zmíněné práci [1].

### 3.4.2 Popis podprogramů

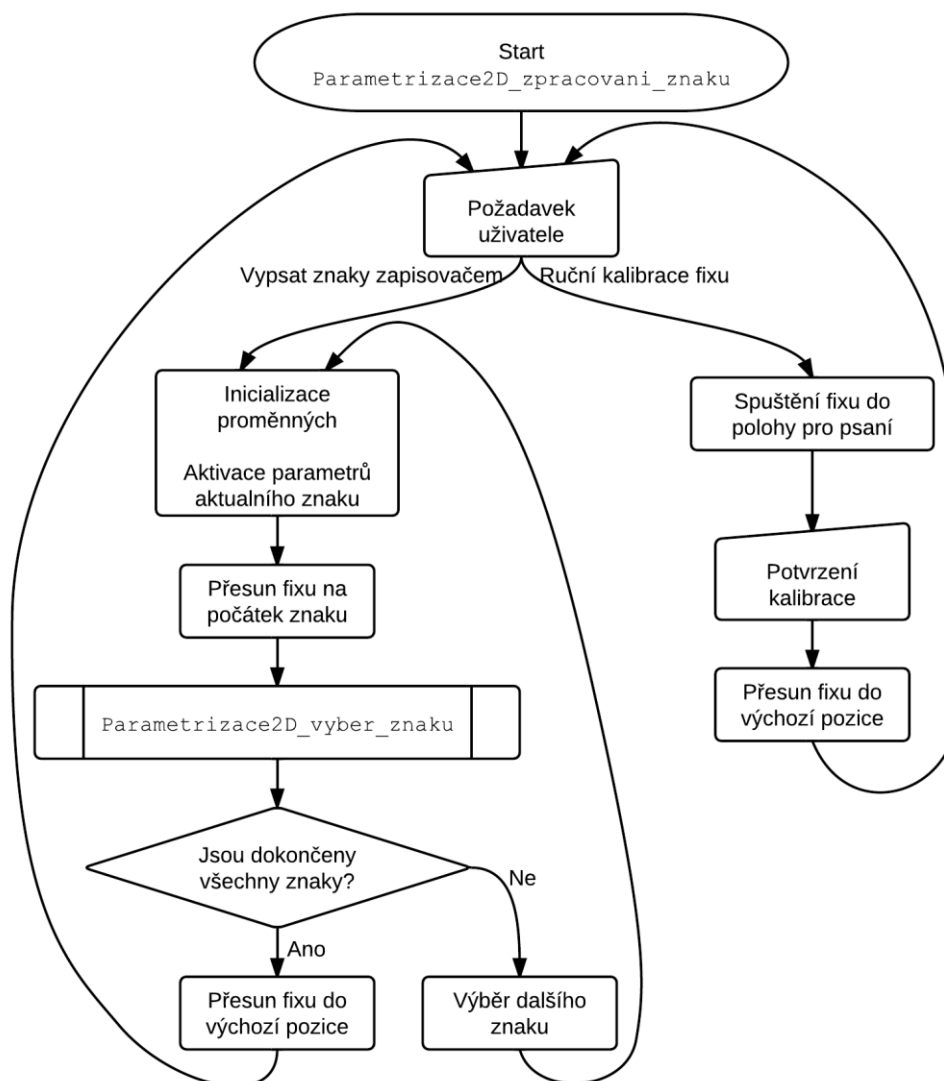
`Parametrizace2D_main`

Je naprogramován v Ladder diagramu.

Stisknutím tlačítka č. 0 dojde k vynulování kroků, proměnné `pom_STOP` a k nastavení `pom_INIT`. Tím se spustí motory os. Jakmile jsou všechny motory spuštěny, `pom_INIT` se vynuluje a osy najíždí do pozice Home. Po jejím dosažení se `STEP` nastaví na 1 a osy se začnou přesouvat do výchozí polohy. Až této polohy dosáhnou, `STEP` se nastaví na 2 a začne se zpracovávat podprogram `Parametrizace2D_zpracovani_znaku`.

Tento podprogram se rovněž stará o bezpečnost. Pokud sepne optická závora, bezpečnostní tlačítko nebo tlačítko č. 2, nastaví se `pom_STOP`, dojde k vypnutí os a signalizační LED č. 1 se rozsvítí. Jestliže nějaká bezpečnostní funkce vypne osy, je potřeba tento stav potvrdit tlačítkem č. 4. Tím se ukončí signalizace (LED č. 1 zhasne), `pom_STOP` se vynuluje a motory os se opět zapnou.

## Parametrizace2D\_zpracovani\_znaku



Obrázek 3.10: Vývojový diagram - zpracování znaku

Je vytvořen pomocí sekvenčního funkčního diagramu.

Po jeho aktivaci čeká na povel z ovládacího formuláře v MS Word. Ten může spustit větev pro výpis znaků, nebo pro kalibraci hrotu fixu.

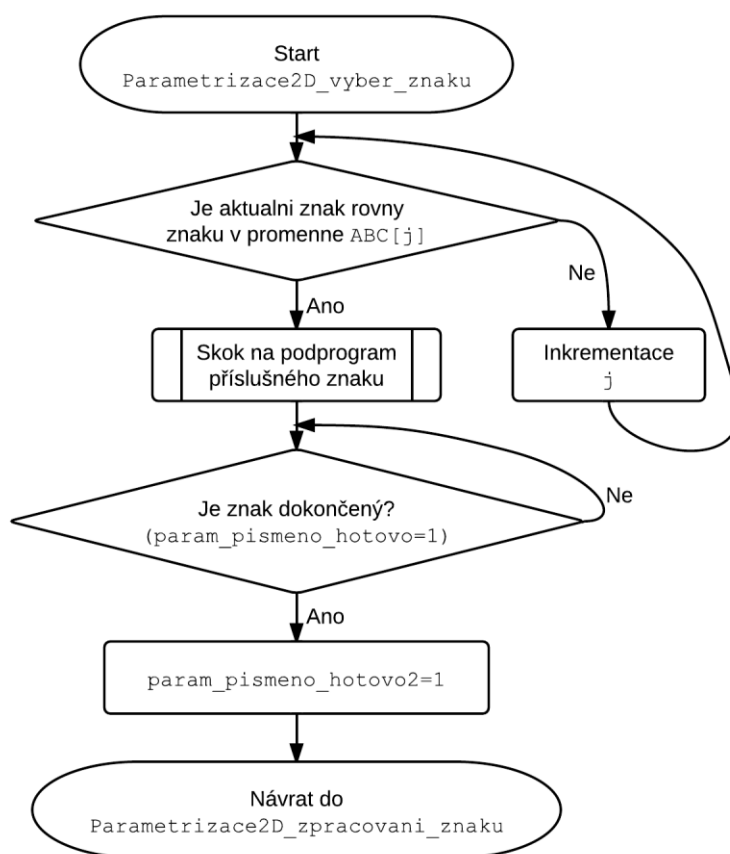
Větev pro výpis znaků pokračuje inicializací proměnných, naplněním ZOOM (velikost znaku), diakri (ASCII hodnota znaku) a pole param\_presun\_pozice (pozice počátku znaku) hodnotami pro konkrétní znak z pole poleznak (na pozici i). Následuje přesun osy na počátek znaku. Po jeho dosažení se spustí podprogram Parametrizace2D\_vyber\_znaku.



Jakmile je znak dokončen (`param_pismo_hotovo2` je nastaven), ověří se, zda byly vypsány všechny znaky (`i=param_pocet_znaku-1`). Pokud ano, osy se přesunou do výchozí pozice a celý program se vrátí zpět na začátek, to znamená, že čeká na povel z ovládacího formuláře. V opačném případě se inkrementuje `i` a program pokračuje znovu vypsáním dalšího znaku.

Požaduje-li uživatel kalibraci znaku, spustí se druhá větev. Ta spustí osu na úroveň, při které probíhá psaní, a čeká na uživatelské potvrzení. Následně se vrátí zpět do výchozí polohy a program se přesune na začátek.

Parametrizace2D\_vyber\_znaku



Obrázek 3.11: Vývojový diagram - výběr znaku

Je naprogramován ve strukturovaném textu.

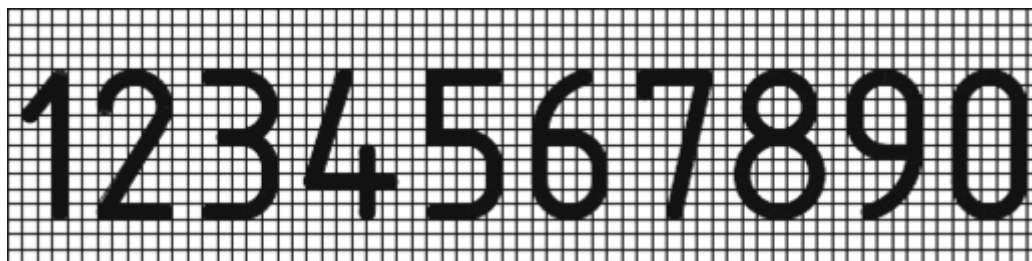
Porovnává ASCII hodnotu vypisovaného znaku `poleznak[i].znak` s ASCII hodnotami všech vytvořených znaků uložených v poli `ABC[]`. Jakmile se tyto hodnoty budou rovnat, zavolá se podprogram na vypsání příslušného znaku.

Podprogram znaku, po svém dokončení, nastaví proměnnou param\_pismeno\_hotovo. Pokud je toto splněno, Parametrizace2D\_vyber\_znaku nastaví param\_pismeno\_hotovo2 a dojde k návratu z podprogramu.

cislo\_0\_nula

Podprogram pro znak číslice 0 i ostatních číslic (1-9) je obdobný, jako již vytvořené znaky písmen viz bakalářská práce p. Poláka [1]. Rozdíl je pouze v tom, že se na konci nevolá podprogram k vypsání mezery a že se nastavuje proměnná param\_pismeno\_hotovo.

Číslice jsou tvořeny souřadnicemi, které se musí ke každému znaku uložit. Vzor technického písma pro určení souřadnic je na Obrázek 3.12.



Obrázek 3.12: Vzor technického písma - číslice

## 4 ZÁVĚR

Zadáním mé bakalářské práce bylo navázat na předchozí práce realizované na modelu 2D zapisovače a vytvořit aplikaci pro PC, která jej bude úkolovat pomocí moderních komunikačních prostředků. Další úkol představoval realizaci technického písma (číslic), návrh dalších možností ovládání a ověření funkčnosti systému.

V první fázi jsem se seznámil s instrumentací firmy Rockwell Automation a předchozími bakalářskými pracemi. Po jejich prostudování jsem navrhl dvě možnosti parametrizování, a to ve vizualizačním prostředí FactoryTalkView MachineEdition a v MS Excel. V případě vizualizačního prostředí se k datům může přistupovat pomocí komunikace vytvořené v RSLinx. MS Excel musí využívat DDE, nebo OPC spojení.

Při studování těchto možností přišlo také v úvahu vytvořit aplikaci pomocí MS Word. Ten umožňuje tvořit programy ve VBA a komunikovat pomocí DDE stejně jako MS Excel. Navíc v něm lze pohodlně zadávat text přímo do textového editoru, měnit velikost znaků a jejich umístění na stránce. Jedinou nevýhodou je, že znakům nelze elegantním způsobem přidělovat natočení. Tuto nevýhodu jsem oproti zmíněným výhodám nepovažoval za podstatnou a aplikaci jsem tedy realizoval v MS Word. Pomocí VBA jsem vytvořil ovládací formulář a podprogramy, jejichž úkolem je uživatele informovat o průběhu programu, zamezit mu vkládání nepovolených znaků a nežádoucí úpravy dokumentu. Dále se s jejich pomocí kalibruje poloha fixu a přenáší se data do řídicího systému.

V RSLogix 5000 jsem doplnil stávající programy p. Poláka [1] o programy číslic a o programy, které umožňují vypisování zadaných znaků.

Funkčnost celé aplikace pro parametrizování 2D zapisovače jsem demonstroval vypsáním ukázkové stránky. Pro porovnání jsem tutéž stránku vytiskl přímo z MS Word. Obě jsou uvedeny, jako přílohy.

Pokud by se měl tento projekt dále rozvíjet, navrhl bych například optimalizovat bezpečnostní funkci optické závory, vytvořit komplexní aplikaci pro ovládání celého zapisovače nebo vylepšit práci s motory (vypínání při nečinnosti, změna rychlosti).

# LITERATURA

[1] POLÁK, J. *Písmo pro 3D zapisovač*. Brno : Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. str. 42, Bakalářská práce. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D.

[2] MATĚJÍČEK, M. *Řízení 3 os manipulátoru*. Brno : Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. str. 61, Bakalářská práce. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D..

[3] CHLAD, P. *Vizualizace manuálního ovládání modelu manipulátoru*. Brno : Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. str. 45, Bakalářská práce. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D..

[4] ROCKWELL AUTOMATION. SKUPINA SERVOPOHONŮ KINETIX. *MOTION-BR005A-CS-P*. [Online] September 2007. [Citace: 27. 12. 2011.] Dostupné z: [http://literature.rockwellautomation.com/idc/groups/literature/documents/br/motion-br005\\_-cs-p.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/br/motion-br005_-cs-p.pdf).

[5] ROCKWELL AUTOMATION. 1756 ControlLogix Integrated Motion Modules, SERCOS Motion Catalog. *1756-TD004B-EN-E*. [Online] May 2010. [Citace: 27. 12. 2011.] Dostupné z: [http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1756-td004\\_-en-e.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1756-td004_-en-e.pdf).

[6] ROCKWELL AUTOMATION. 1756 ControlLogix Controllers Specifications. *1756-td001\_-en-e*. [Online] May 2011. [Citace: 8. 1. 2012.] Dostupné z: [http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1756-td001\\_-en-e.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1756-td001_-en-e.pdf).

- [7] ROCKWELL AUTOMATION. RSLinx Classic Getting Results Guide. *linx-gr001\_-en-e*. [Online] November 2011. [Citace: 8. 1. 2012.] Dostupné z: [http://literature.rockwellautomation.com/idc/groups/literature/documents/gr/linx-gr001\\_-en-e.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/gr/linx-gr001_-en-e.pdf).
- [8] FOXON. OPC Server / OPC Klient. *opc-server-opc-klient-c-72*. [Online] [Citace: 27. 12. 2011.] Dostupné z: <http://www.foxon.cz/opc-server-opc-klient-c-72.html>.
- [9] FOXON. OPCEx - OPC Add-in Klient pro MS Excel. *opcex-opc-addin-klient-pro-ms-excel-p-614*. [Online] 2012. [Citace: 8. 1. 2012.] Dostupné z: <http://www.foxon.cz/opcex-opc-addin-klient-pro-ms-excel-p-614.html>.
- [10] ROCKWELL AUTOMATION. FactoryTalk View Machine Edition User's Guide. *VIEWME-UM004H-EN-E*. [Online] June 2011. [Citace: 27. 12. 2011.] Dostupné z: [http://literature.rockwellautomation.com/idc/groups/literature/documents/um/viewme-um004\\_-en-e.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/um/viewme-um004_-en-e.pdf).
- [11] SCHOP, J. Connecting Excel to ControlLogix. *connecting\_controllogix\_excel*. [Online] 2009. [Citace: 6. 1. 2012.] Dostupné z: [http://www.plcdev.com/connecting\\_controllogix\\_excel](http://www.plcdev.com/connecting_controllogix_excel).
- [12] ROCKWELL AUTOMATION. Logix5000 Controllers Design Considerations Reference Manual. *1756-rm094\_-en-p*. [Online] December 2011. [Citace: 7. 1. 2012.] Dostupné z: [http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm094\\_-en-p.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm094_-en-p.pdf).

# **SEZNAM PŘÍLOH**

## **A) PŘÍLOŽENÉ CD**

- A.1) Program v RSLogix 5000
- A.2) Exportované tagy z RSLogix 5000
- A.3) Celkový report programu v RSLogix 5000
- A.4) Program v MS Word
- A.5) Font osifont
- A.6) Bakalářská práce Jakub Heczko

## **B) UKÁZKOVÉ STRÁNKY**

- B.1) Stránka vypsána zapisovačem
- B.2) Předloha z aplikace MS Word